Application and System Memory Use, Configuration, and Problems on Bassi

Richard Gerber

Lawrence Berkeley National Laboratory

NERSC User Services

ScicomP 13, Garching, Germany, July 17, 2007

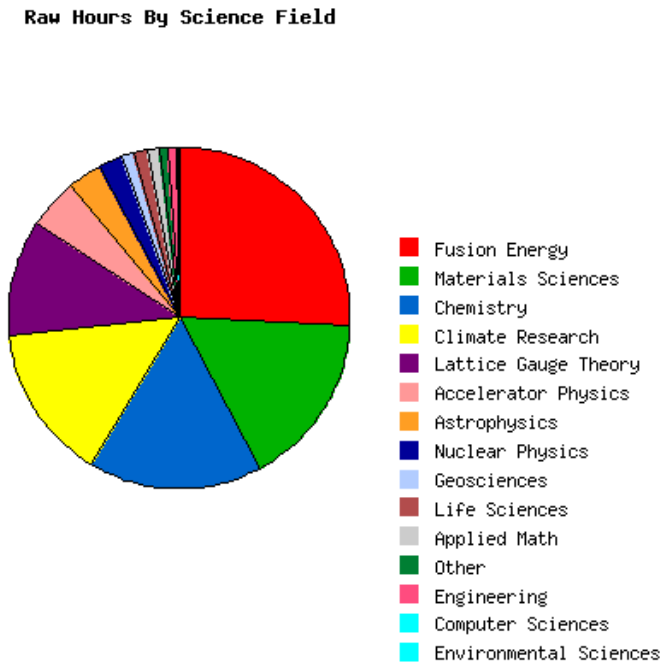
NERSC is supported by the Office of Advanced Scientific Computing Research in the Department of Energy Office of Science under contract number DE-AC02-05CH11231.

Overview

- About Bassi
- Memory on Bassi
- Large Page Memory (It's Great!) 
- System Configuration
- Large Page "Gotchas" 
- The Plague 
- Workload Characterization

Bassi Description

- **NERSC IBM POWER 5 p575: Bassi**
  - **111 (114) node single-core 1.9 GHz P5**
  - **8-way SMP**
  - **32 GB physical memory per node**
  - **Very diverse workload**
    - **~400 active users**
    - **~400 jobs per day**
    - **28% node-hrs >32 nodes**
    - **44% node-hrs < 8 nodes**
    - **12% node-hrs = 1 node**
    - **Fortran, C, C++, mixed-mode**
    - **MPI, OpenMP, pThreads**
    - **shmget()**
    - **MPMD, SPMD, emb. parallel**



Memory on Bassi

NERSC is supported by the Office of Advanced Scientific Computing Research in the Department of Energy Office of Science under contract number DE-AC02-05CH11231.

Bassi Memory Overview

- Each node has 32 GB of memory
- Memory is partitioned into two types
  - Large Page pool
  - Small Page pool
- Large pages are required for HPS
- AIX uses small pages only
- Applications can use either
  - Small pages only
  - Large and small pages

Large Page Memory

It's Great!

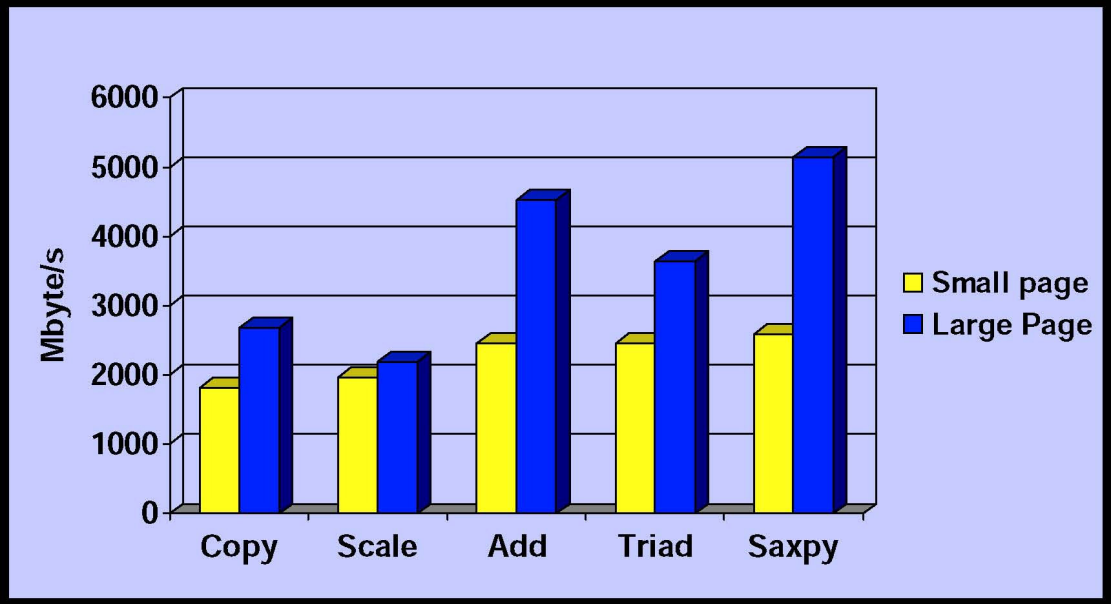
NERSC is supported by the Office of Advanced Scientific Computing Research in the Department of Energy Office of Science under contract number DE-AC02-05CH11231.

Large Page Memory

- Large memory pages are 16 MB
  - Can not be swapped to disk
  - Number of large pages per node is set at boot
  - Large Page memory is backed by Small Pages
- Small memory pages are 4 KB
  - Can be swapped to disk
- Large Page memory is good for most scientific applications
  - Enhanced memory bandwidth
  - 16 GB TLB coverage (vs. 4 MB for small)
  - 131072 cache lines (vs. 32 for small)

Memory Bandwidth

Large Pages: Bandwidth Enhancement



POWER5 1.45 GHz

LP Memory & NERSC Benchmarks

- **Large page memory improves performance on NERSC "Bassi" benchmarks**
  - **NPB MG 2.4 Class C: 38%**
  - **NPB SP 2.4 Class C: 16%**
  - **NPB FT 2.4 Class C: 13%**
  - **GTC (PIC Fusion): 7%**
  - **PARATEC (Materials): 8%**
  - **CAM 3.0 (Climate Atms): 1%**

System Configuration

NERSC is supported by the Office of Advanced Scientific Computing Research in the Department of Energy Office of Science under contract number DE-AC02-05CH11231.

System Configuration

- It appears that Large Pages are good for HPC applications
- Why not configure system for as many large pages as possible, leaving adequate small pages for AIX?
- After consultation with IBM, NERSC chose to allocate 24 GB to the large page pool

System and Application Memory

- On an idle compute node
  - AIX uses about 4 GB of small page memory; ~4 GB free
  - HPS reserves 69 large pages, or 1.078 GB; 22.922 GB free
- Applications can access the remaining memory
  - 4 GB + 23 GB ≅ 27 GB

Large Page “Gotchas”

NERSC is supported by the Office of Advanced Scientific Computing Research in the Department of Energy Office of Science under contract number DE-AC02-05CH11231.

Large Page "Gotchas"

- Users must enable codes to run in Large Pages; it is not the default
- The application stack must reside in small page memory
- FORTRAN 90 "regular" arrays are allocated on the stack by default (? –qlargepage)
- Shared memory segments are allocated in small page memory by default
- OpenMP PRIVATE data is allocated on the stack (? –qsmallstack)
- Large Page memory allocation is slow (scripts and serial commands should use small pages)

Bad Things Happen When Small Page Memory is Exhausted

- Small Page memory must swap to disk when exhausted
- When jobs exhaust small-page memory
  - Slows, hangs, or kills the application
  - Makes the node unresponsive
  - GPFS dies
  - Causes other havoc
- Why?
  - Theory: AIX memory manager can't deal with 8 tasks concurrently trying to allocate/access large chunk of memory that is not physically present and must be paged to disk

NERSC Mitigation Efforts

- Set the mp* compilers to enable large pages by default
  - "Serial" compiled codes use small pages
- Set runtime environment variables to force batch jobs into large pages
- Identify shmget() programmers and tell them to set SHM_LGPAGE and SHM_PIN flags
- Tell FORTRAN 90 users to use –qsave to force "normal" arrays into static (LP?) memory
  - -qlargepage ?
- WLM ConsumableMemory settings and low paging space kills

The Plague

NERSC is supported by the Office of Advanced Scientific Computing Research in the Department of Energy Office of Science under contract number DE-AC02-05CH11231.

The Memory Exhaustion Plague

- **NERSC has been plagued by users exhausting small page memory and effectively disabling nodes**
  - **Node becomes unresponsive; services die**
  - **Users' jobs die**
  - **Node may or may not recover by itself**
  - **System admins get paged**
  - **Consultants have to contact users to try to get more information**
  - **Users get disabled so they won't kill the nodes again**

System Monitoring

- We starting monitoring the memory usage on all compute nodes
  - Used LoadLeveler “llstatus –l” utility
  - Free small page memory
  - Free large page memory
  - Sampled every 15 mins since mid April 2007
  - Recorded user and StepID running on each node

Monitoring Results

- In about 70 days we found
  - Large pages exhausted 3.3% of the time
  - Small pages exhausted 0.87% (1/115) of the time
  - Node paging to disk 0.29% (1/345) of the time
  - Paging with LP use, but free LPs 0.24% (1/417) of the time
- We identified three common node failure modes
  - Just using too much memory (>27 GB)
  - Running without enabling large page use
  - Using some large pages, but nonetheless exhausting small page memory
- We contacted users to get more information about their codes and job scripts

Causes

- Users override default programming environment
  - Executables are not large-page enabled
  - Batch environmental variable not set (LDR_CNTRL)
  - ‘bash’ configuration file issues
- Using large automatic arrays and OpenMP PRIVATE data (?)
- Making shared memory calls without LP flags
- Programming errors
- Users don’t really understand their code’s memory requirements
- Third-party libraries
- Unknown issues

Solutions

- Work one-on-one with users to resolve known issues
- Work with third-party developers to incorporate P5-friendly code
  - Global Arrays, NWChem, MOLPRO, GAMESS
- NERSC reduced Large Page pool to 20 GB on 7/11/2007 to accommodate codes that need a larger stack
- Possibly set –qsave as default for f90 compilers? Or –qlargepage? –qsmallstack?
- New system configuration setting promised by IBM to monitor and kill jobs based on small page memory use

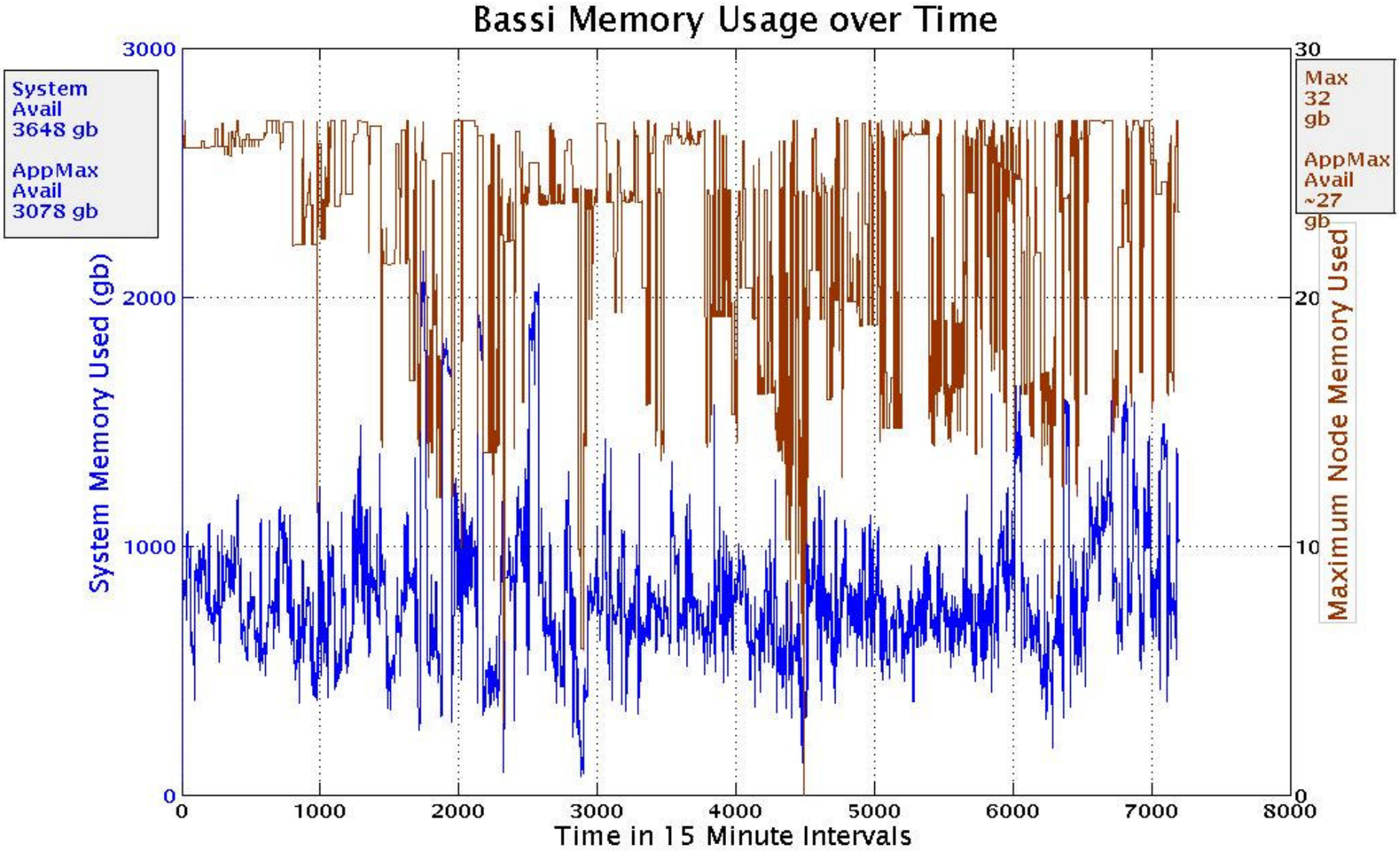
Workload Characterization

NERSC is supported by the Office of Advanced Scientific Computing Research in the Department of Energy Office of Science under contract number DE-AC02-05CH11231.

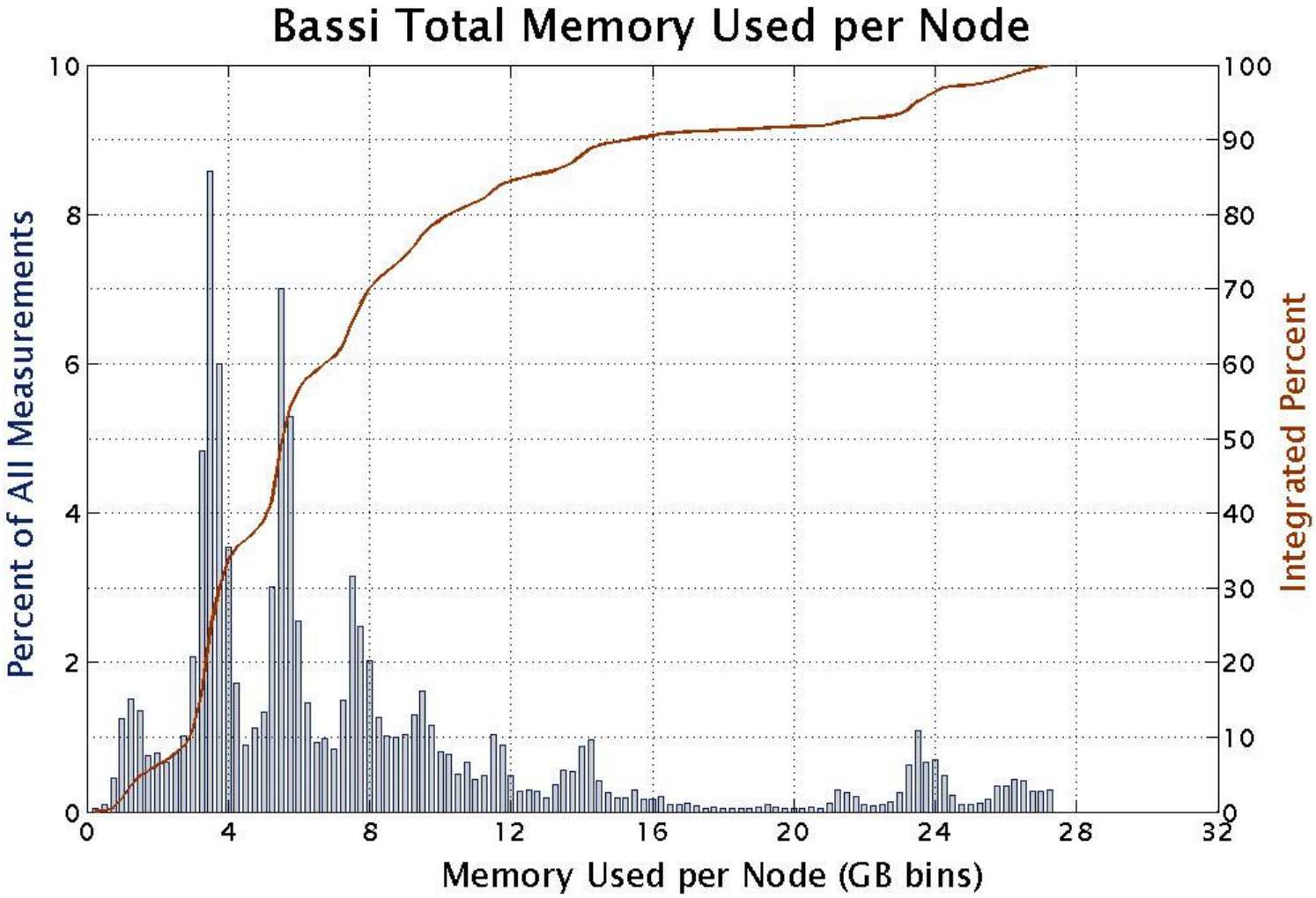
Workload Characterization

- **We have these memory snapshots, so we can ask questions about the NERSC workload**
  - **How much memory is used on average?**
  - **What is the maximum memory usage?**
  - **How many Large Pages should we allocate?**
  - **Can we help users understand their codes' memory use patterns (e.g. find memory leaks)**
  - **Can we identify classes of jobs based on memory use patterns?**

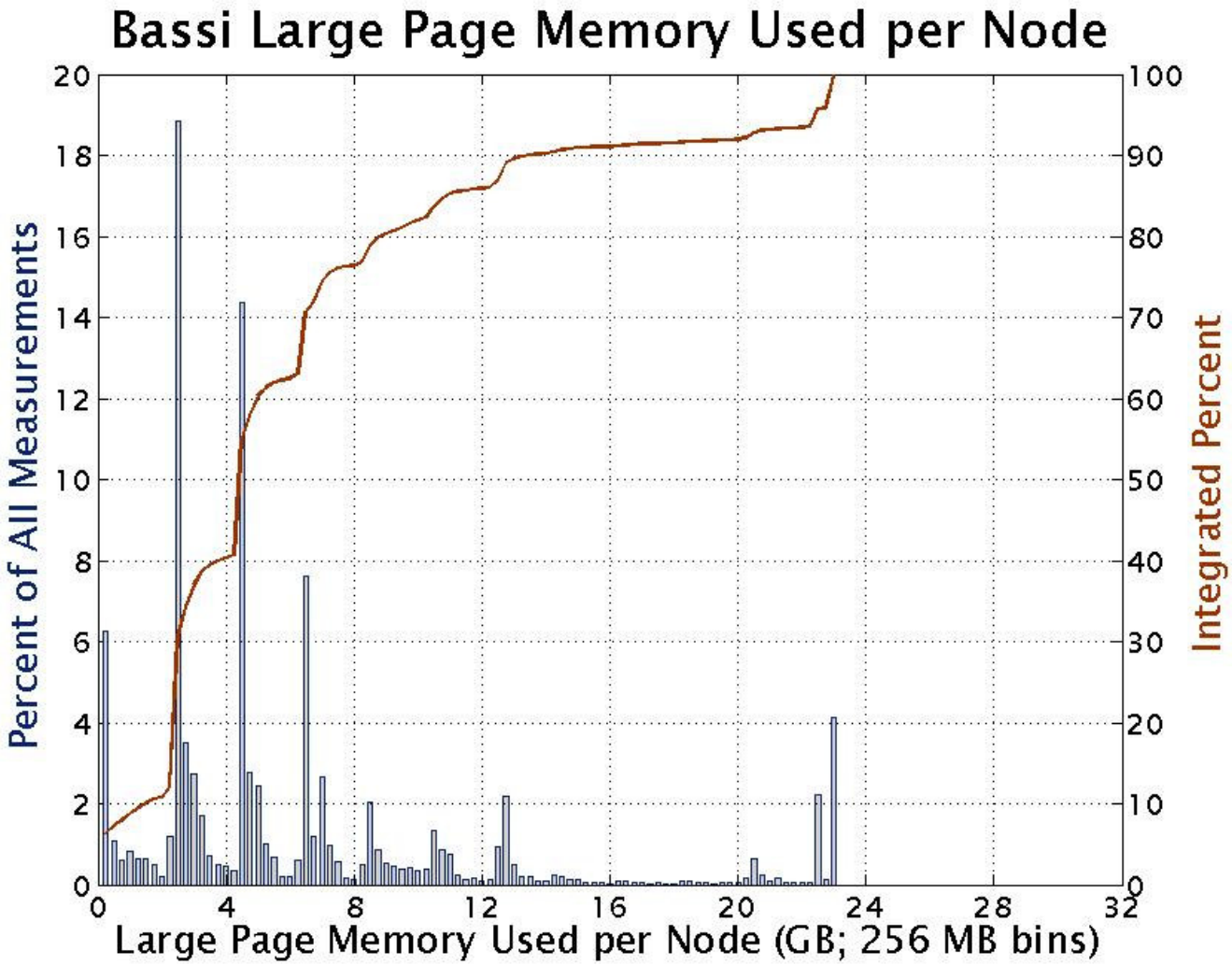
Total and Max Memory Used



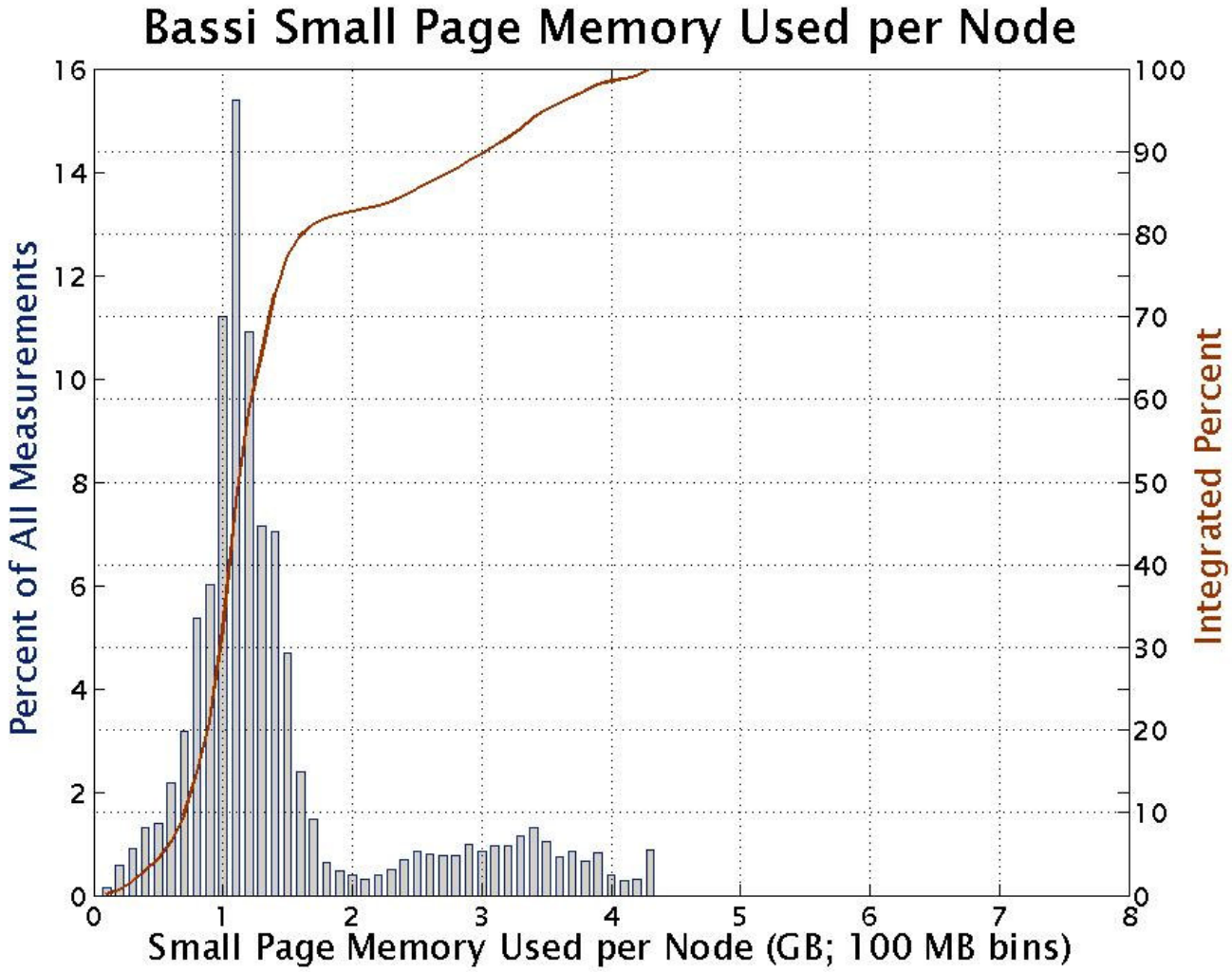
Total Memory Used



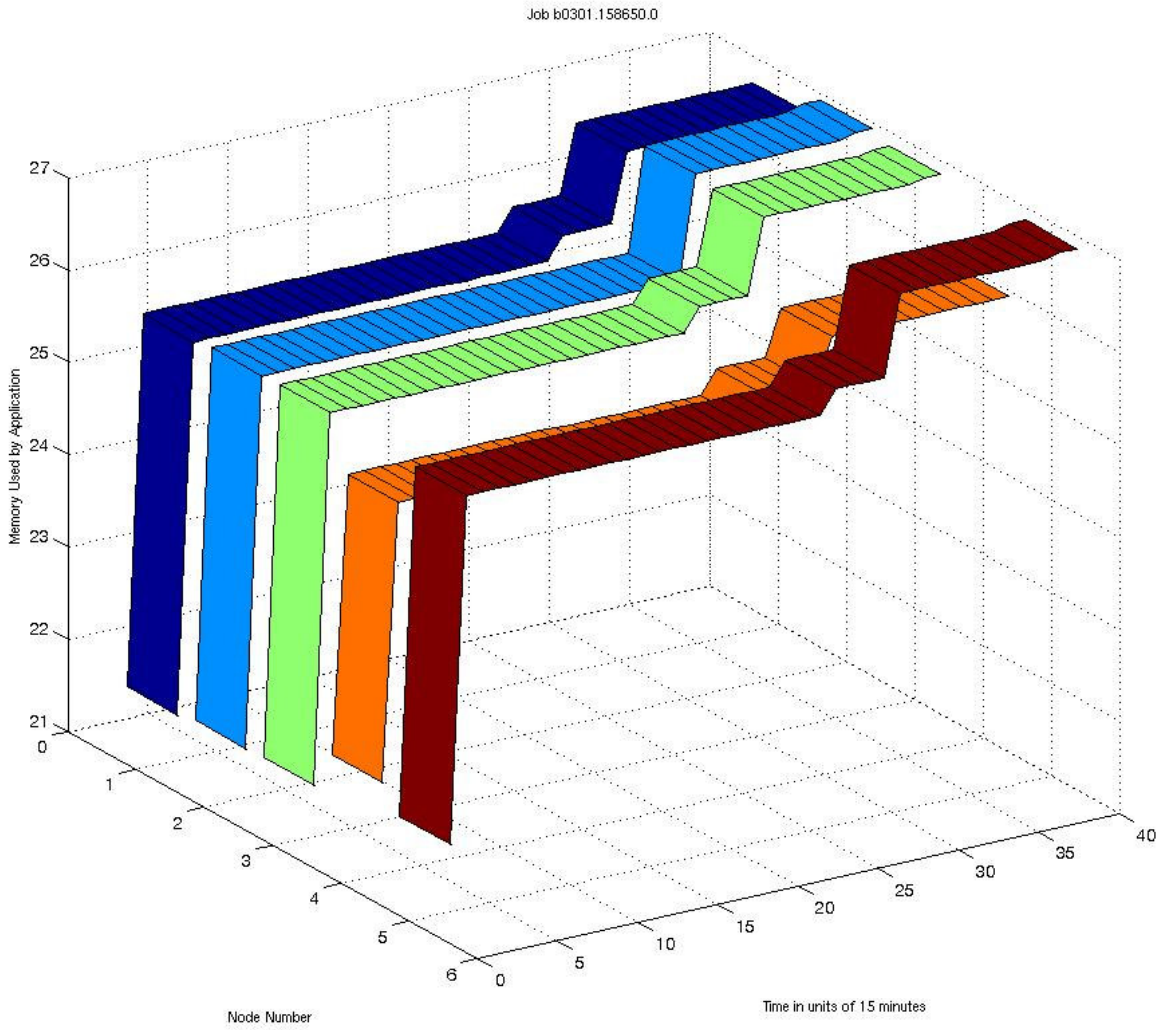
Large Page Memory Used



Small Page Memory Used



Job Characterization



Office of Science
U.S. DEPARTMENT OF ENERGY

BERKELEY LAB